Ensemble Clustering with Logic Rules

Deniz Akdemir
Department of Plant Breeding & Genetics
Cornell University
Ithaca, NY

August 24, 2012

Abstract

In this article, the logic rule ensembles approach to supervised learning is applied to the unsupervised or semi-supervised clustering. Logic rules which were obtained by combining simple conjunctive rules are used to partition the input space and an ensemble of these rules is used to define a similarity matrix. Similarity partitioning is used to partition the data in an hierarchical manner. We have used internal and external measures of cluster validity to evaluate the quality of clusterings or to identify the number of clusters.

Keywords & Phrases: ensemble learning, clustering, biological annotation, logic rule, random projection

1 Introduction

Cluster analysis is used to identify groupings of individuals in populations based on data. The identified groups partition the data space and are called clusters. The lack of a universal definition of a cluster, and its task or data dependent nature has resulted in publication of a very large number of clustering algorithms.

The approach taken in this paper for clustering is closely related to the decision tree, rule and logic rule ensemble approaches to the supervised learning problem. The logic rule ensembles combine the propositions about individual input variables with with logic operators "and", "or" and "not" and use them as input variables ([1]) to estimate the target variable. Each rule takes the values in $\{0, 1\}$ and partitions the sample space into two parts. We use the data as input variables and random projections of this data ([17], [2]) as target variables and combine the rules learned from these supervised problems into one clustering using the cluster based similarity partitioning framework of Ghosh et. al. ([21]).

In the following section, we review how the importance sampling learning ensemble (ISLE) algorithm ([12]) can be used to generate an ensemble of conjunctive rules and the more familiar and concise logic rules for the supervised learning problem. This approach is then adopted to the clustering problem in

Section 3. In Section 4, we illustrate the ensemble clustering algorithm and compare it to several popular clustering algorithms.

2 Supervised Learning with (Logic) Rules

Given a learning task and a relevant data set, we can generate a set of models from a predetermined model family. Bagging bootstraps the training data set [3] and produces a model for each bootstrap sample. Random forest ([14, 5]) creates a diverse set of models by randomly selecting a few aspects of the data set while generating each model. AdaBoost [10] and ARCing [4] iteratively build models by varying case weights (up-weighting cases with large current errors and down-weighting those accurately estimated) and employs the weighted sum of the estimates of the sequence of models. There have been few attempts to unify these ensemble learning methods. One such framework is the ISLE due to Popescu & Friedman [11].

We are to produce a regression model to predict the continuous outcome variable y from p vector of input variables $\boldsymbol{x}$. We will generate models from a given model family $\mathscr{F} = \{f(\boldsymbol{x}, \theta) : \theta \in \Theta\}$ indexed by the parameter θ. The final ensemble models considered by the ISLE framework have an additive form:

$$F(\boldsymbol{x}) = w_0 + \sum_{j=1}^{M} w_j f(\boldsymbol{x}, \theta_j) \qquad (1)$$

where $\{f(\boldsymbol{x}, \theta_j)\}_{j=1}^M$ are base learners selected from $\mathscr{F}$. ISLE uses a two-step approach to produce $F(\boldsymbol{x})$. The first step involves sampling the space of possible models to obtain $\{\widehat{\theta}_j\}_{j=1}^M$. The second step proceeds with combining the base learners by choosing weights $\{w_j\}_{j=0}^M$ in (1).

The pseudo code to produce M models $\{f(\boldsymbol{x}, \widehat{\theta}_j)\}_{j=1}^M$ under ISLE framework is given below:

Algorithm 2.1: ISLE(M, ν, η)

$F_0(\boldsymbol{x}) = 0.$
for j=1 **to** M

$$\textbf{do} \begin{cases} (\widehat{c}_j, \widehat{\theta}_j) = \underset{(c,\theta)}{\operatorname{argmin}} \sum_{i \in S_j(\eta)} L(y_i, F_{j-1}(\boldsymbol{x}_i) + c f(\boldsymbol{x}_i, \theta)) \\ T_j(\boldsymbol{x}) = f(\boldsymbol{x}, \widehat{\theta}_j) \\ F_j(\boldsymbol{x}) = F_{j-1}(\boldsymbol{x}) + \nu \widehat{c}_j T_j(\boldsymbol{x}) \end{cases}$$

return $(\{T_j(\boldsymbol{x})\}_{j=1}^M$ and $F_M(\boldsymbol{x}).)$

Here $L(.,.)$ is a loss function, $S_j(\eta)$ is a subset of the indices $\{1, 2, \ldots, n\}$ chosen by a sampling scheme η, $0 \le \nu \le 1$ is a memory parameter.

The classic ensemble methods of Bagging, Random Forest, AdaBoost, and Gradient Boosting are special cases of ISLE ensemble model generation procedure [20]. In Bagging and Random Forests the weights in 1 are set to predetermined values, i.e. $w_0 = 0$ and $w_j = \frac{1}{M}$ for $j = 1, 2, \ldots, M$. Boosting calculates

these weights in a sequential fashion at each step by having positive memory ν, estimating c_j and takes $F_M(\boldsymbol{x})$ as the final prediction model.

Friedman & Popescu [11] recommend learning the weights $\{w_j\}_{j=0}^{M}$ using lasso [22]. Let $T = (T_j(\boldsymbol{x}_i))_{i=1}^{n}{}_{m=1}^{M}$ be the $n \times M$ matrix of predictions for the n observations by the M models in an ensemble. The weights $(w_0, \boldsymbol{w} = \{w_m\}_{m=0}^{M})$ are obtained from

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}(\boldsymbol{y} - w_0\mathbf{1}_n - T\boldsymbol{w})'(\boldsymbol{y} - w_0\mathbf{1}_n - T\boldsymbol{w}) + \lambda\sum_{m=1}^{M}|w_m|. \tag{2}$$

$\lambda > 0$ is the shrinkage operator, larger values of λ decreases the number of models included in the final prediction model. The final ensemble model is given by

$$F(\boldsymbol{x}) = w_0 + \sum_{m=1}^{M} w_m T_m(\boldsymbol{x}). \tag{3}$$

Given a set of decision trees, rules can be extracted from each of these trees to produce a collection of rules. Let $R = (r_k(\boldsymbol{x}_i))_{i=1}^{n}{}_{k=1}^{K}$ be the $n \times K$ matrix of rules for the n observations by the K rules in the ensemble. The **rulefit** algorithm of Friedman & Popescu [12] uses the weights $(w_0, \boldsymbol{w} = \{w_k\}_{k=0}^{K})$ that are estimated from

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}(\boldsymbol{y} - w_0\mathbf{1}_n - R\boldsymbol{w})'(\boldsymbol{y} - w_0\mathbf{1}_n - R\boldsymbol{w}) + \lambda\sum_{k=1}^{K}|w_k| \tag{4}$$

in the final prediction model

$$F(\boldsymbol{x}) = w_0 + \sum_{k=1}^{K} w_k r_k(\boldsymbol{x}). \tag{5}$$

A conjunctive rule $r(\boldsymbol{x}) = \prod_{l=1}^{p} I(x_l \in s_l)$ can also be expressed as a logic rule (also called Boolean expressions and logic statement) involving only the $\wedge$ ("and") operator. In general, a logic statement is constructed using the operators $\wedge$ ("and"), $\vee$ ("or") and ^{c} ("not") and brackets. An example simple logic rule is

$$l(\boldsymbol{x}) = [I(x_1 \in s_1) \vee I^c(x_2 \in s_2)] \wedge I(x_3 \in s_3).$$

It should be noted that the representation of a logic rule in general is not unique. However, it can be shown that all logic rules can be expressed in disjunctive normal form where we only use $\vee$ combinations of $\wedge$ terms. Coupling this with the De Morgan's laws we see that all logic rules can actually be written as conjunctive rules. However, the additional "or" operator in logic rules in disjunctive normal form provide consolidation of interchangeable rules and therefore provide more precise and interpretable results.

Given the logic rules $\{l_1(\boldsymbol{x}), \{l_2(\boldsymbol{x}), \ldots, \{l_L(\boldsymbol{x})\}$, let $S = (l_k(\boldsymbol{x}_i))_{i=1}^{n}{}_{k=1}^{L}$ be the $n \times K$ matrix of logic rules for the n observations by the L logic rules in the

ensemble. We propose using the weights $(w_0, \boldsymbol{w} = \{w_k\}_{k=0}^{L})$ that are estimated from

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} (\boldsymbol{y} - w_0 \mathbf{1}_n - S\boldsymbol{w})'(\boldsymbol{y} - w_0 \mathbf{1}_n - S\boldsymbol{w}) + \lambda \sum_{k=1}^{L} |w_k| \tag{6}$$

in the final prediction model

$$F(\boldsymbol{x}) = w_0 + \sum_{k=1}^{L} w_k l_k(\boldsymbol{x}). \tag{7}$$

We call this the logic rule ensemble.

The absolute values of the standardized coefficients

$$\{Imp_k = |w_k|\sqrt{v_k(1 - v_k)}\}, k = 1, 2, \ldots, L$$

can be used to evaluate the importance of a rule ([12]). Here v_k is the support of the rule k and defined as $v_k = \sum_{i=1}^{n} s_k(\boldsymbol{x}_i)/n$. A measure of importance for each variable can be obtained as the sum the importances of rules that involve that variable.

3 (Logic) Rule Ensemble Clustering

The main difference between the supervised learning and unsupervised learning is the existence of a target variable in the former. The sample partitioning approaches discussed in the previous section partition the sample space into clusters. Each rule defines a clustering of the sample space into two components which give a good segregation of the target variable. When we are not provided with a target variable, we construct our target variables by mapping the input variables. Each of these target variables can be used to extract several interesting rules and overall cluster rules are obtained from combining the rules for many target variables into an ensemble distance matrix. The details of the procedure are described below.

Algorithm 3.1: SS-ISCA(X, Y, M, m, ν)

R_1 : A random projection of Y
for $j = 1$ **to** M

$$\textbf{do} \begin{cases} \text{Generate m logic rules } \{l_\ell(\boldsymbol{x})\}_{\ell=1}^{m} \text{ to estimate } R_j \text{ from } X\text{:} \\ \qquad S_j(\boldsymbol{x}) \Leftarrow \{l_\ell(\boldsymbol{x})\}_{\ell=1}^{m} \\ T(X) \Leftarrow \{S_j(\boldsymbol{x}_i)\}_{i=1}^{n}{}_{d=1}^{j} \\ R_{j+1} : \text{A random projection of Y} \\ R_{j+1} \Leftarrow (I - \nu P_{T(X)}) R_{j+1} \end{cases}$$

return $(T(X))$

Here $P_{T(X)}$ is the projection matrix on to the space spanned by the columns of $T(X)$ calculated as $T(X)[T(X)'T(X)]^{-}T(X)$. The time it takes to calculate

this matrix grows quickly as the algorithm advances in its stages. For high dimensional datasets, where M or m should be selected large, this makes the ISCA algorithm to be slow. One simple strategy is to apply lasso regression to select rules at each stage j and therefore reducing m. Another strategy is to sample the columns of $T(X)$ to calculate $P_{T(X)}$ at each stage, and in this case ν is used as the sampling proportion. We have used both of these techniques successfully for high dimensional clustering problems. Of course, if the memory parameter ν is set to zero then we do not have to worry about calculating $P_{T(X)}$. In this case the ISCA algorithm is parallelizable.

From $T(X)_{n\times r}$, a similarity matrix can be obtained as $S(X) = T(X)T(X)'/r$. Note that $T(X)$ is a sparse matrix. The ijth element of $S(X)$ is the percentage of times the ith and the jth observations fire the same rules. From the similarity matrix $S(X)$, we calculate a distance matrix $D(X)$. The clustering of the observations is accomplished by applying a distance based hierarchical clustering algorithm ([16]) to the rule based distance matrix $D(X)$. The method we have used to combine the clusters from many rules is referred to as the cluster based similarity partitioning in Ghosh et. al. ([21]).

In distance based hierarchical clustering, first, each object is assigned to its own cluster. At each consecutive stage the two most similar clusters are joined until there is a single cluster. The distances between clusters are recalculated at each stage by a linkage criterion such as single-linkage, complete linkage or average linkage. In our illustrations in the following section, we have uniformly used the average linkage criterion for combining clusters.

For generation of rules useful for ensemble clustering when there are no target variables, we modify the semi-supervised rule generation algorithm in 3.1 as follows:

Algorithm 3.2: ISCA(X, M, m, ν)

R_1 : A random projection of X
for $j = 1$ **to** M

$$\textbf{do} \begin{cases} \text{Generate m logic rules } \{l_\ell(\boldsymbol{x})\}_{\ell=1}^m \text{ to estimate } R_j \text{ from } X\text{:} \\ \qquad S_j(\boldsymbol{x}) \Leftarrow \{l_\ell(\boldsymbol{x})\}_{\ell=1}^m \\ T(X) \Leftarrow \{S_j(\boldsymbol{x}_i)\}_{i=1}^n{}_{d=1}^{j} \\ R_{j+1} : \text{A random projection of X} \\ R_{j+1} \Leftarrow (I - \nu P_{T(X)})R_{j+1} \end{cases}$$

return $(T(X))$

Hierarchical clustering produces nested clusterings of the data set. In order to determine the final clustering, the number of clusters has to be determined. When the clustering problem is accompanied by external class labels or external benchmarks and the number of clusters is known, the quality of a cluster can be measured by Rand measure (R) ([18]), Jaccard index (J) ([7]), Fowlkes Mallows index (FM) ([9]), Wallace indices (W01, W10), etc... Otherwise internal clustering quality measures like Silhouette (silhouette) ([19]), Dunn Index (dunn) ([8]), Connectivity (connect) ([6]) can be used to identify the number

of clusters. These measures can also be used to compare the quality of several clusterings. The article by Handl et al. ([13]) provides an excellent overview of cluster validation measures.

4 Illustrations

We illustrate the clustering approaches in four real data sets. In addition to the SS-ISCA and ISCA approaches, we employ existing methods like model based clustering (Mclust), partitioning around medoids (PAM), divisive analysis clustering (DIANA) and random forest clustering (RF). The different clusterings are compared using the internal measures (Silhouette, Dunn index, Connectivity) or using the external measures (Rand measure, Fowlkes Mallows index, Wallace indices, Jaccard index). In the case where a supervisory target variable was available and there were only two clusters, we also provided the p values from comparing the means for the target variable in these clusters.

Example 4.1. *(Fisher's Iris Data Set) The results of clustering the Fisher's Iris data set are displayed in Table 1. The existing data labels are used to calculate the internal validity measures. We have also provided internal measures of cluster quality. ISCA algorithm is uniformly the best according to the internal measures. With respect to the external validation measures ISCA algorithm ranks second after the model based clustering.*

Table 1: (Fisher's Iris Data Set, Unsupervised Clustering) Internal and external measures of cluster validity. ISCA algorithm is uniformly the best or second best according to these measures. ** and * are used to mark the best and the second best clusterings correspondingly.

	Internal			**External**				
	silhouette	**dunn**	**connect**	**R**	**FM**	**W01**	**W10**	**J**
ISCA	0.55**	0.12**	7.40**	0.89*	0.83*	0.86*	0.81*	0.71*
RF	0.48	0.03	23.26	0.71	0.66	0.79	0.54	0.47
PAM	0.55**	0.10	10.09*	0.88	0.82	0.84	0.81	0.70
DIANA	0.54	0.11*	12.43	0.86	0.80	0.81	0.78	0.66
Mclust	0.50	0.07	14.18	0.96**	0.94**	0.94**	0.93**	0.88**

Example 4.2. *(FHB Data Set, Semi-Supervised Clustering) FHB is a plant disease caused by the fungus Fusarium Graminearum and results in tremendous losses by reducing grain yield and quality. In addition to the decrease in grain yield and quality, another damage due to FHB is the contamination of the crop with mycotoxins. Therefore, breeding for improved FHB resistance is an important breeding goal. The Fusarium Head Blight (FHB) data set contains information on* 2251 *markers, along with the FHB and DON levels for* 622 *elite barley lines. The data is available from the author upon request. A very detailed explanation of this data set is given in [15]. We would like to segregate the*

622 barley lines into two groups, low resistance and high resistance lines. The results from clustering this data using different approaches are summarized by the box plots in Figure 1. SS-ISCA clearly gives the best segregation of the FHB variable among other clusterings which we measure by the p value corresponding to the two sample t test for comparing group means. Some internal measures for cluster quality for clusterings by different clustering approaches are provided in Table 2.

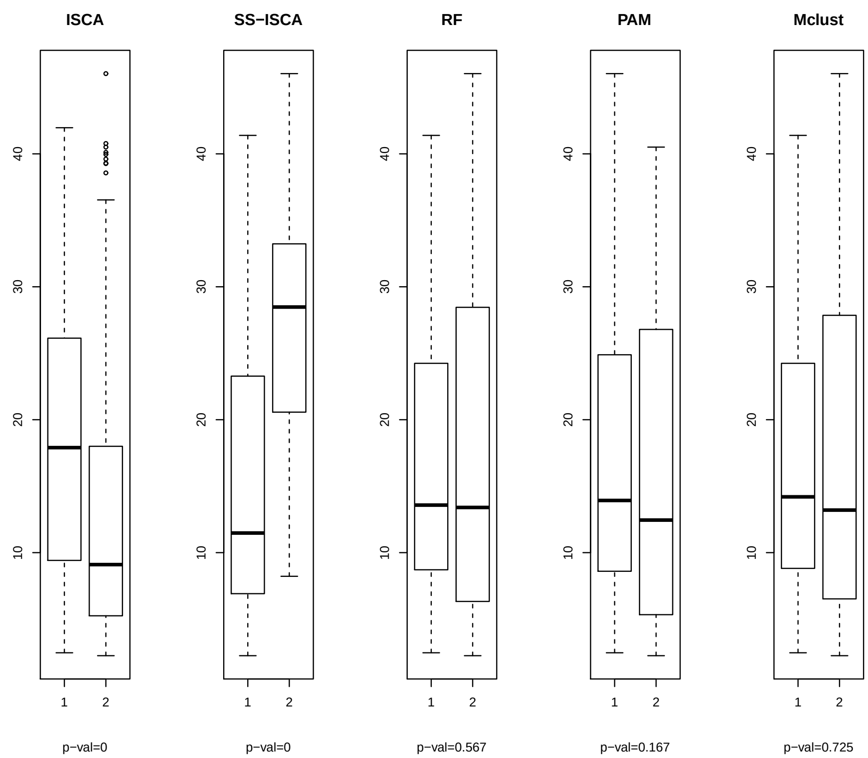

Figure 1: (FHB Data Set, Semi-Supervised Clustering) The FHB data set contains information on 2251 markers, along with the FHB and DON levels for 622 elite barley lines. p values from the t tests corresponding to different clustering approaches indicate that the SS-ISCA and ISCA produce groups that are different from each other in terms of the mean FHB.

Example 4.3. *(Stem Rust Data Set) The stem rusts is a disease affecting cereal crops. Crop species which are affected by the disease include wheat, barley and triticale. We had estimated breeding values of stem rust resistance for* 374 *lines of wheat. In addition* 1624 *markers were available for these lines. The box plots in Figure 2 compare the stem rust resistance for the groups from several clustering algorithms. The p-values from the two sample t test are also provided. The best segregation is obtained again by the ISCA approach.*

Table 2: (FHB Data Set, Semi-Supervised Clustering) SS-ISCA and ISCA clusterings outperform other clusterings.

	silhouette	**dunn**	**connect**
SS-ISCA	0.108	0.419**	44.450**
ISCA	0.114*	0.344*	67.763*
RF	0.092	0.344*	111.410
PAM	0.126**	0.160	206.743
Mclust	0.114*	0.317	126.918

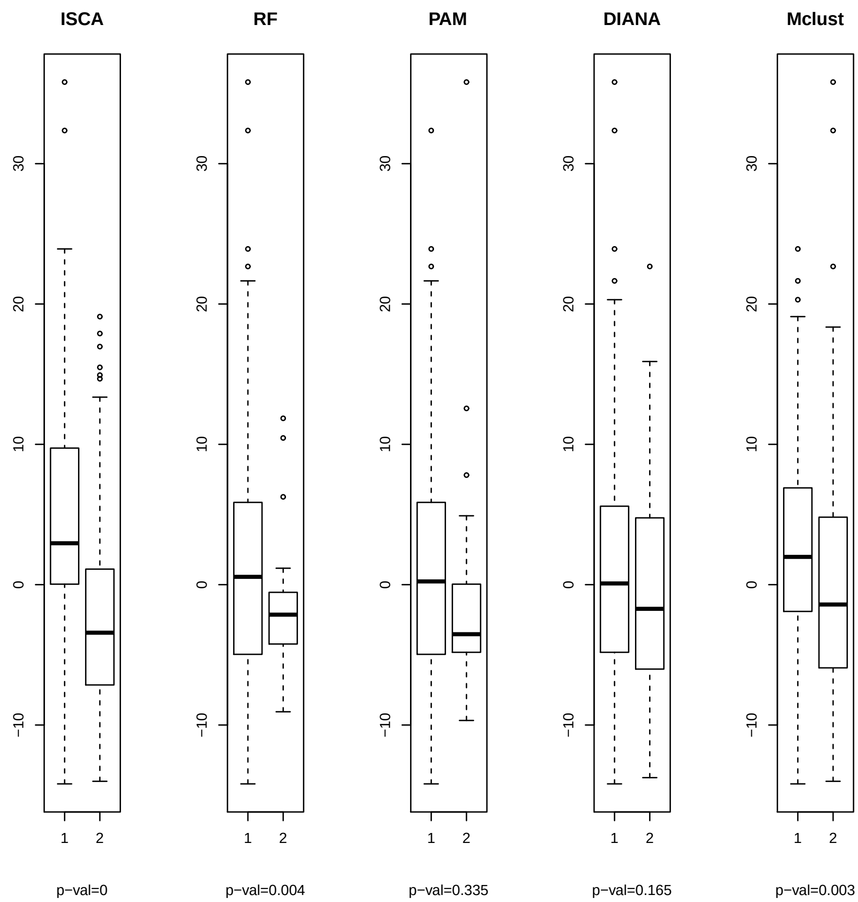

Figure 2: (Stem Rust Data Set, Semi-Supervised Clustering) The best segregation is obtained by ISCA approach.

Example 4.4. *(Mouse Data Set, Learning the Number of Clusters) We use data from an Afyymetrix microarray experiment comparing gene expression of mesenchymal cells from two distinct families, neural crest and mesoderm derived. The dataset consists of* 147 *genes and expressed sequence tags (EST) which were determined to be significantly differentially expressed between the two cell groups. For further description of the dataset and the experiments the reader is referred to Bhattacherjee et al. (2007). The internal measures of cluster quality is displayed for* 2 *to* 30 *groups clustering by ISCA in Table 3. The optimal number of clusters is determined to be two using the connectivity and silhouette width. Although the Dunn Index increases almost uniformly after* 3 *clusters and attains very high levels after* 10 *or more clusters, there is indication that* 2 *groups provides a reasonable clustering.*

5 Conclusion

We have discussed how we can use an ensemble of logic rules for unsupervised and semi-supervised cluster learning. Our examples show that the approaches introduced herein are promising, they produce high quality clusters as indicated by many internal and external measures of cluster quality.

Acknowledgments

I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this article. This research was also supported by the USDA-NIFA-AFRI Triticeae Coordinated Agricultural Project, award number 2011-68002-30029.

References

[1] D. Akdemir. Quantifying gene x gene interactions via logic rule ensembles. *Arxiv preprint arXiv: ?*, 2012.

[2] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.

[3] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[4] L. Breiman. Arcing classifier (with discussion and a rejoinder by the author). *The annals of statistics*, 26(3):801–849, 1998.

[5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

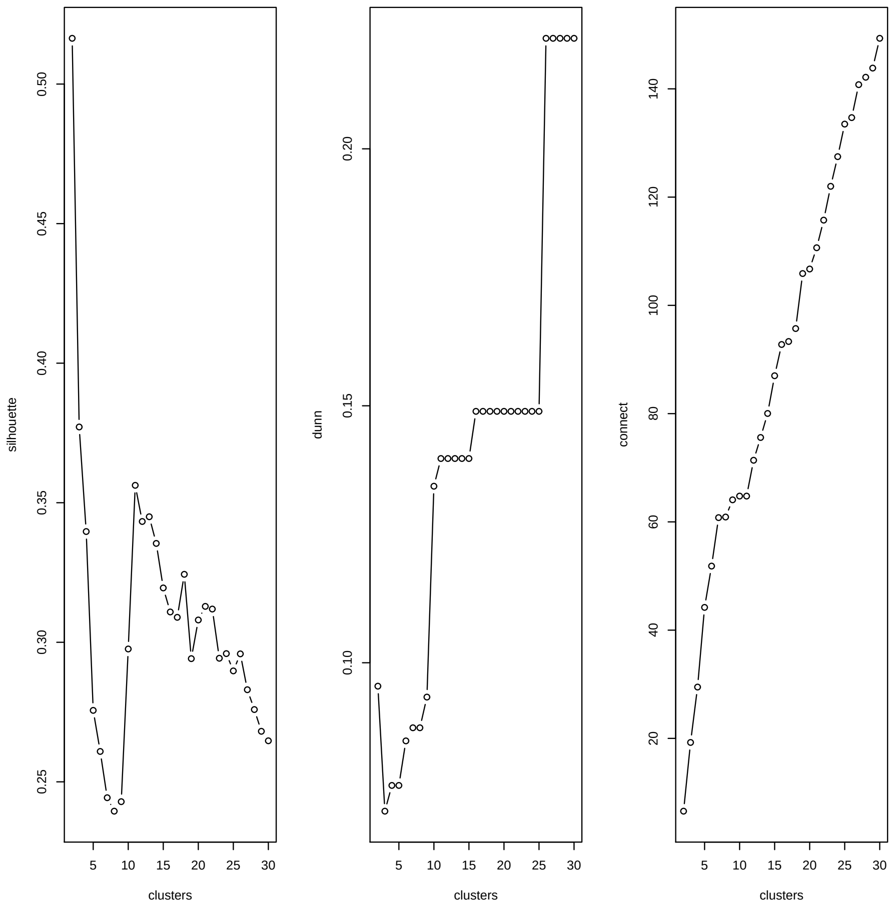

Figure 3: (Mouse Data Set, Learning the Number of Clusters) The internal measures of cluster quality is displayed for 2 to 30 groups clustering by ISCA. Hierarchical clustering with two clusters performs the best in general.

[6] D.L. Davies and D.W. Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227, 1979.

[7] M.P. Dubé. *Étude de la diversité génétique au sein des génomes nucléaire et chloroplastique chez les cinq races connues du Striga gesnerioides, une plante parasite d'importance mondiale*. PhD thesis, Université Laval, 2009.

[8] J.C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[9] E.B. Fowlkes and C.L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, pages 553–569, 1983.

[10] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning-International Workshop then Conference-*, pages 148–156. Morgan Kaufmann Publishers, Inc., 1996.

[11] J.H. Friedman and B.E. Popescu. Importance sampled learning ensembles. *Journal of Machine Learning Research*, 94305, 2003.

[12] J.H. Friedman and B.E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

[13] J. Handl, J. Knowles, and D.B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[14] T.K. Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[15] J.L. Jannink, KP Smith, and AJ Lorenz. Potential and optimization of genomic selection for fusarium head blight resistance in six-row barley. *Crop Science*, 52(4):1609–1621, 2012.

[16] L. Kaufman, P.J. Rousseeuw, et al. *Finding groups in data: an introduction to cluster analysis*, volume 39. Wiley Online Library, 1990.

[17] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Segmentation problems. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 473–482. ACM, 1998.

[18] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, pages 846–850, 1971.

[19] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[20] G. Seni and J.F. Elder. Ensemble methods in data mining: improving accuracy through combining predictions. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–126, 2010.

[21] A. Strehl and J. Ghosh. Cluster ensembles-a knowledge reuse framework for combining partitionings. In *Proceedings of the National Conference on Artificial Intelligence*, pages 93–99. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.

[22] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.